

[INICIAR SESIÓN](#)[NUESTROS PLANES](#)[TODOS LOS CURSOS](#)[FORMACIONES](#)[CURSOS](#)[PARA EMPRESAS](#)[ARTÍCULOS DE TECNOLOGÍA > DEVOPS](#)

Trabajando con caminos y carpetas en el terminal



jonilson.silva

02/12/2021



Esse artigo faz parte da
Formação DevOps

```
MINGW64:/c/Users/Jonilson Sousa

Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~
$ cd C:/Users/Jonilson Sousa/Documents
bash: cd: too many arguments

Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~
$ |
```

En este artículo vamos a entender cómo funcionan los caminos de directorios del sistema operacional y como evitar los tropiezos que podemos enfrentar cuando necesitamos utilizar esos caminos, como, por ejemplo, cuando estamos trabajando con repositorios de **Git**.

¿Porque este asunto?

Aquí en **Alura** tenemos un foro donde las alumnas y alumnos pueden consultar sus dudas, con otros alumnos, alumnas, instructores(as), moderadores(as).

Percibí que, principalmente con quien está comenzando a estudiar DevOps, en su mayoría en los cursos de Git (Tenemos dos cursos: [Git y Github: Controle y comparta su código](#) y [Git y Github: Estrategias de ramificación, Conflictos y Pull Requests](#)), existen dudas recurrentes de errores en los caminos de los directorios, o sea de las carpetas, independientemente del sistema operacional.

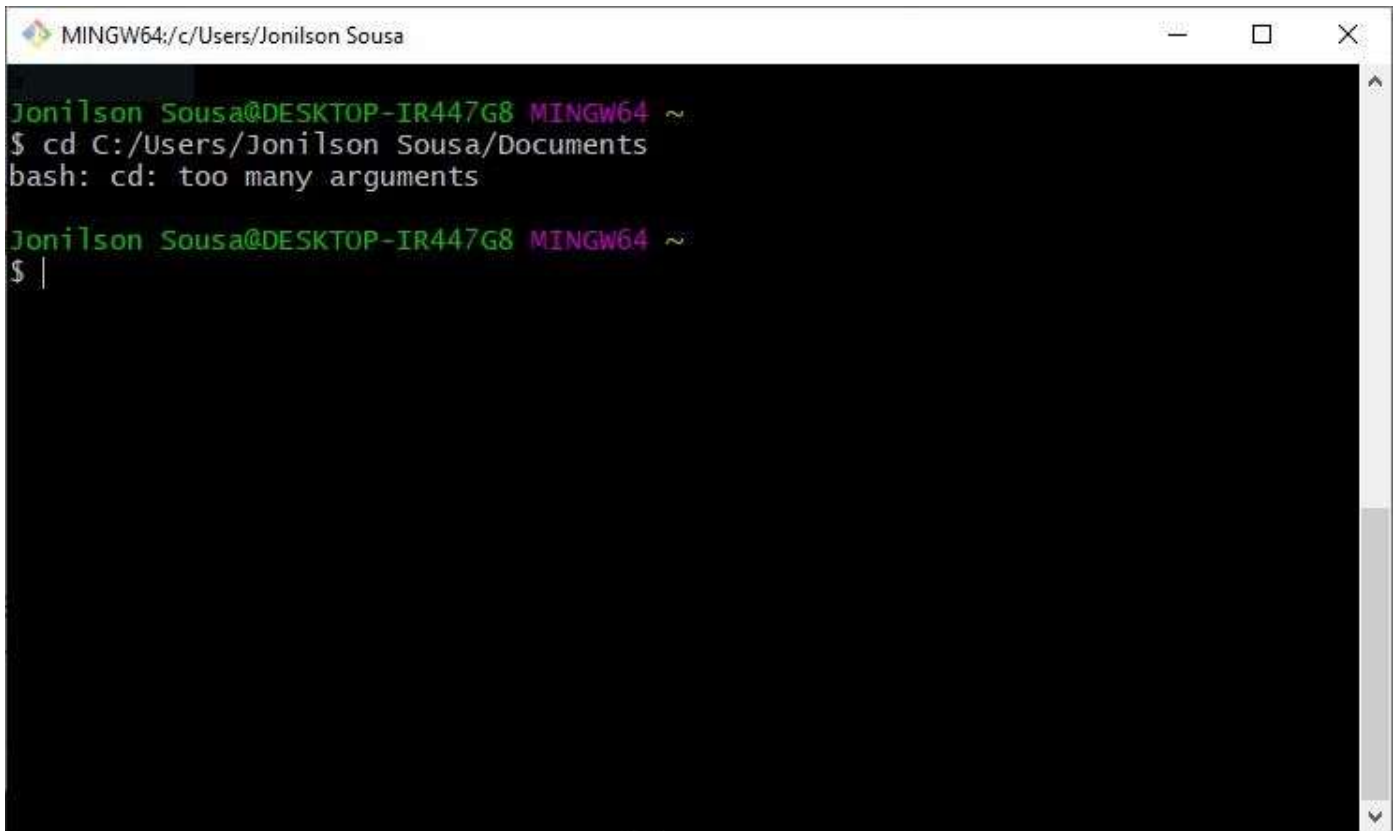
Problemas recurrentes

¿Qué ocurre cuando compramos un computador nuevo o cuando colocamos el sistema operacional Windows (en los sistemas Unix funciona de otra forma) en nuestras máquinas? Colocamos nuestro nombre de usuario compuesto, ¿verdad? Pero, cuando vamos a trabajar con caminos, como es el caso de cuando necesitamos agregar un repositorio remoto al proyecto que estamos trabajando, podemos tener algunos problemas.

Pero no creas que eso solo sucede en esos casos, a veces cuando estamos apenas navegando entre los directorios por medio del terminal tenemos los mismos problemas, imagina que queremos acceder a la carpeta “Documents” que está dentro de la carpeta “Jonilson Sousa” que está dentro de la carpeta “Users”, que por cierto está dentro de la carpeta “C:”, para eso necesitamos utilizar el siguiente comando:

```
cd C:/Users/Jonilson Sousa/Documents
```

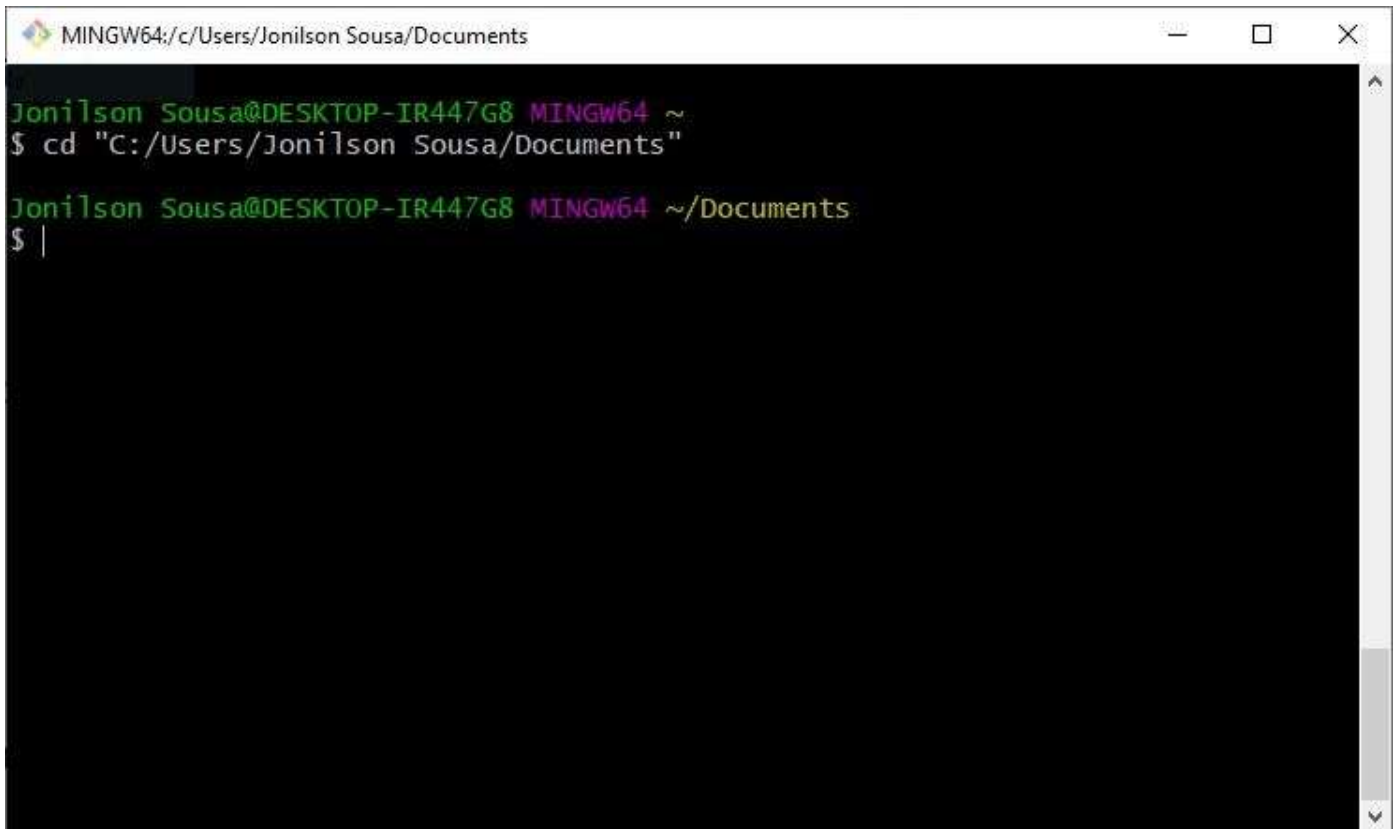
¿Está todo bien con este comando? No en ese caso, vamos a tener error como “too many arguments”, como podemos ver en la imagen de abajo:

A screenshot of a terminal window titled 'MINGW64:/c/Users/Jonilson Sousa'. The terminal shows a prompt 'Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~' followed by the command '\$ cd C:/Users/Jonilson Sousa/Documents'. The output is 'bash: cd: too many arguments'. The prompt is repeated, followed by '\$ |'.

```
MINGW64:/c/Users/Jonilson Sousa
Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~
$ cd C:/Users/Jonilson Sousa/Documents
bash: cd: too many arguments
Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~
$ |
```

Eso sucede porque el terminal no entiende el camino por cuenta del espacio en el nombre del usuario. Eso también puede suceder con otras carpetas. Observe que estamos dentro de la carpeta “Documents” y hay otra carpeta con el nombre “meu trabalho”. Vamos a intentar ejecutar el siguiente comando:

```
cd meu trabalho
```



```
MINGW64:/c/Users/Jonilson Sousa/Documents
Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~
$ cd "C:/Users/Jonilson Sousa/Documents"
Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~/Documents
$ |
```

Recibiremos el mismo error (*"too many arguments"*).

Tu debes estar te preguntando: ¿cómo resolvemos este error? o entonces: ¿será que eso puede suceder en los sistemas basados en Unix, como es el caso de MacOS o Ubuntu, por ejemplo?

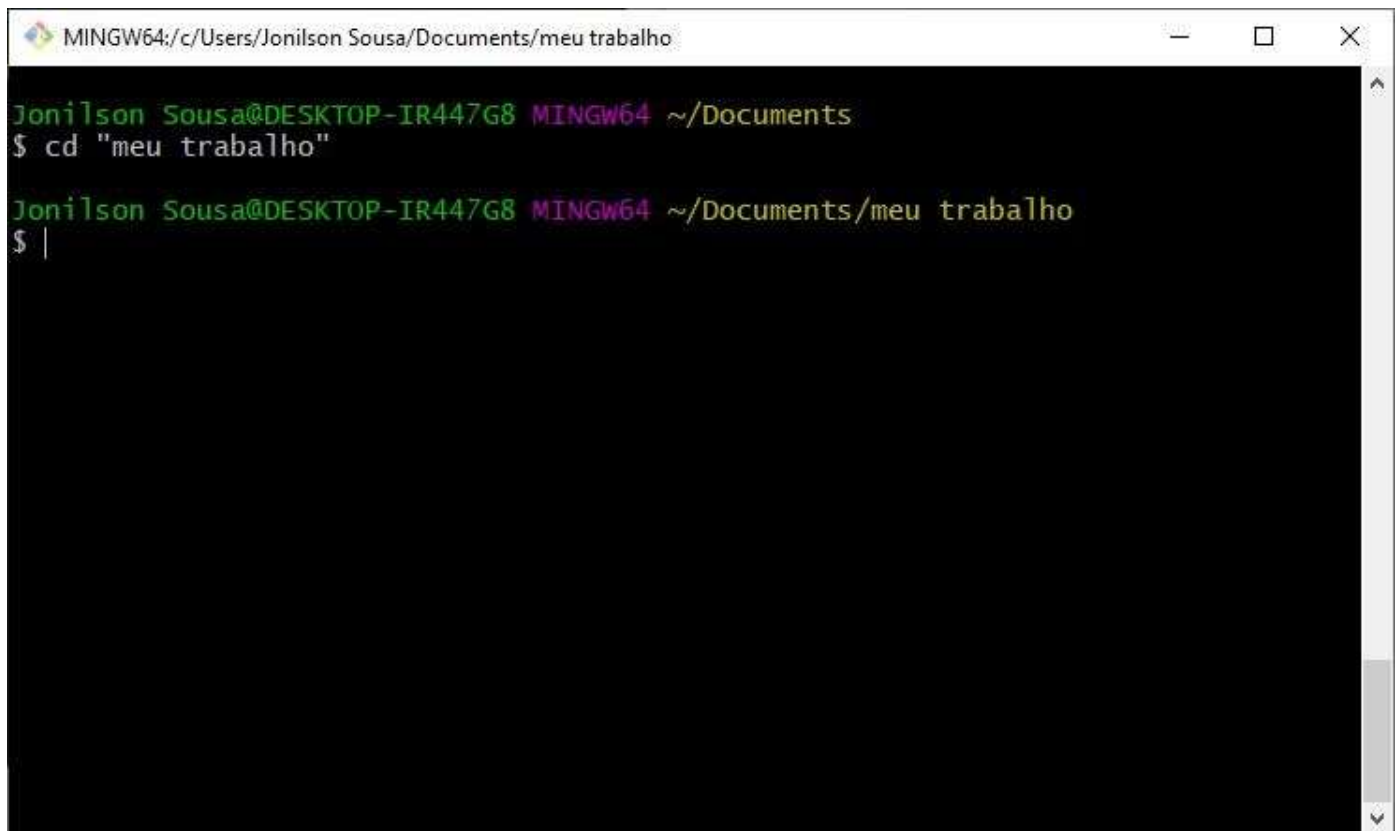
Usuarios en otros sistemas

Los sistemas basados en Unix no permiten que creamos nombres de usuarios utilizando espacios o caracteres especiales. Ya Windows permite. Luego, si fuésemos a crear un usuario en Ubuntu, con el nombre "Jonilson Sousa", (con espacio en el medio), el sistema inmediatamente nos presentaría un mensaje de error.

Pero, si ya tuviéramos el usuario creado (en Windows) con un nombre compuesto, ¿cómo podemos ajustar?

Utilizando nombres compuestos

Podemos contornar esta situación utilizando comillas (" "). Observa los errores anteriores, ahora con comillas:

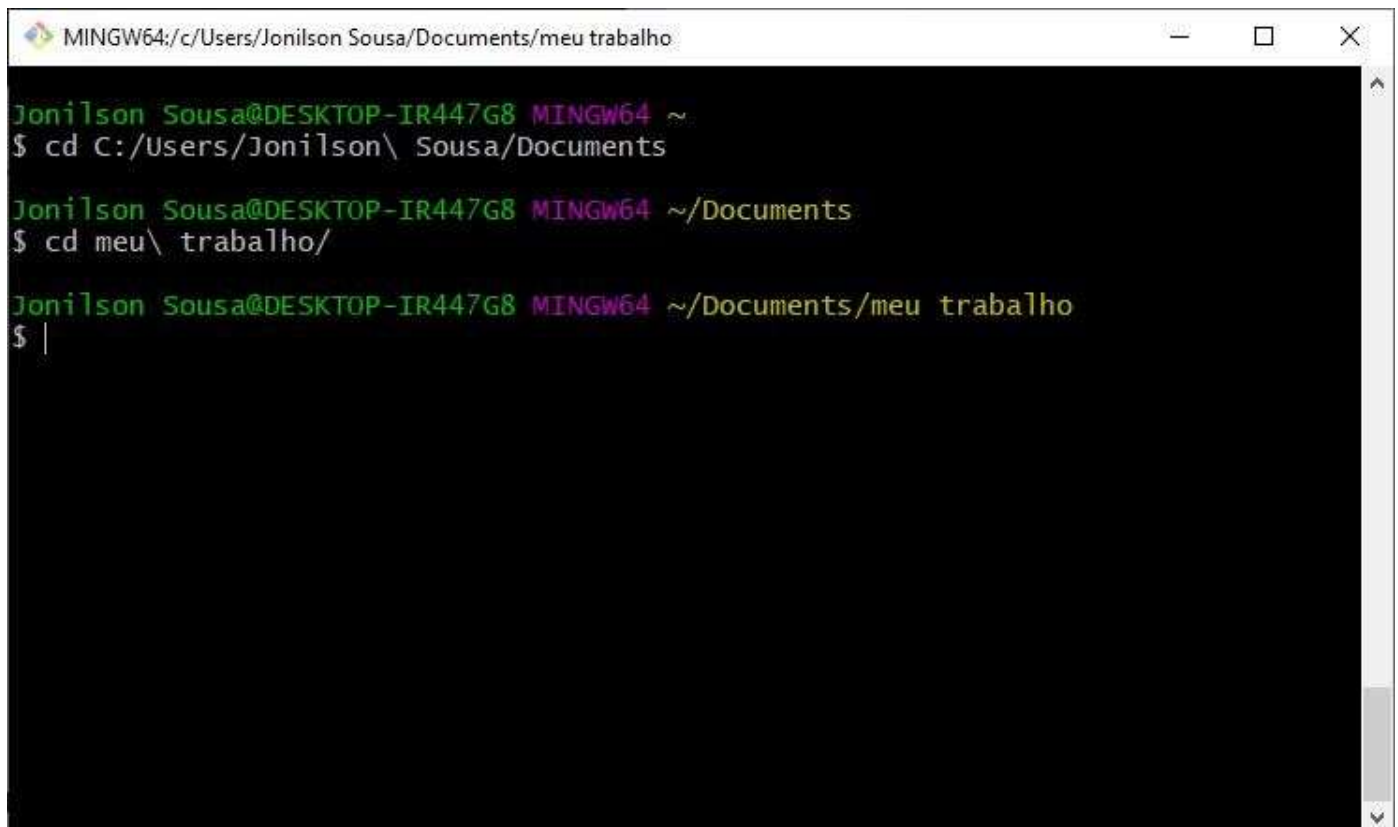


```
MINGW64:/c/Users/Jonilson Sousa/Documents/meu trabalho

Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~/Documents
$ cd "meu trabalho"

Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~/Documents/meu trabalho
$ |
```

Y también accediendo a la carpeta “meu trabalho”:



```
MINGW64:/c/Users/Jonilson Sousa/Documents/meu trabalho

Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~
$ cd C:/Users/Jonilson\ Sousa/Documents

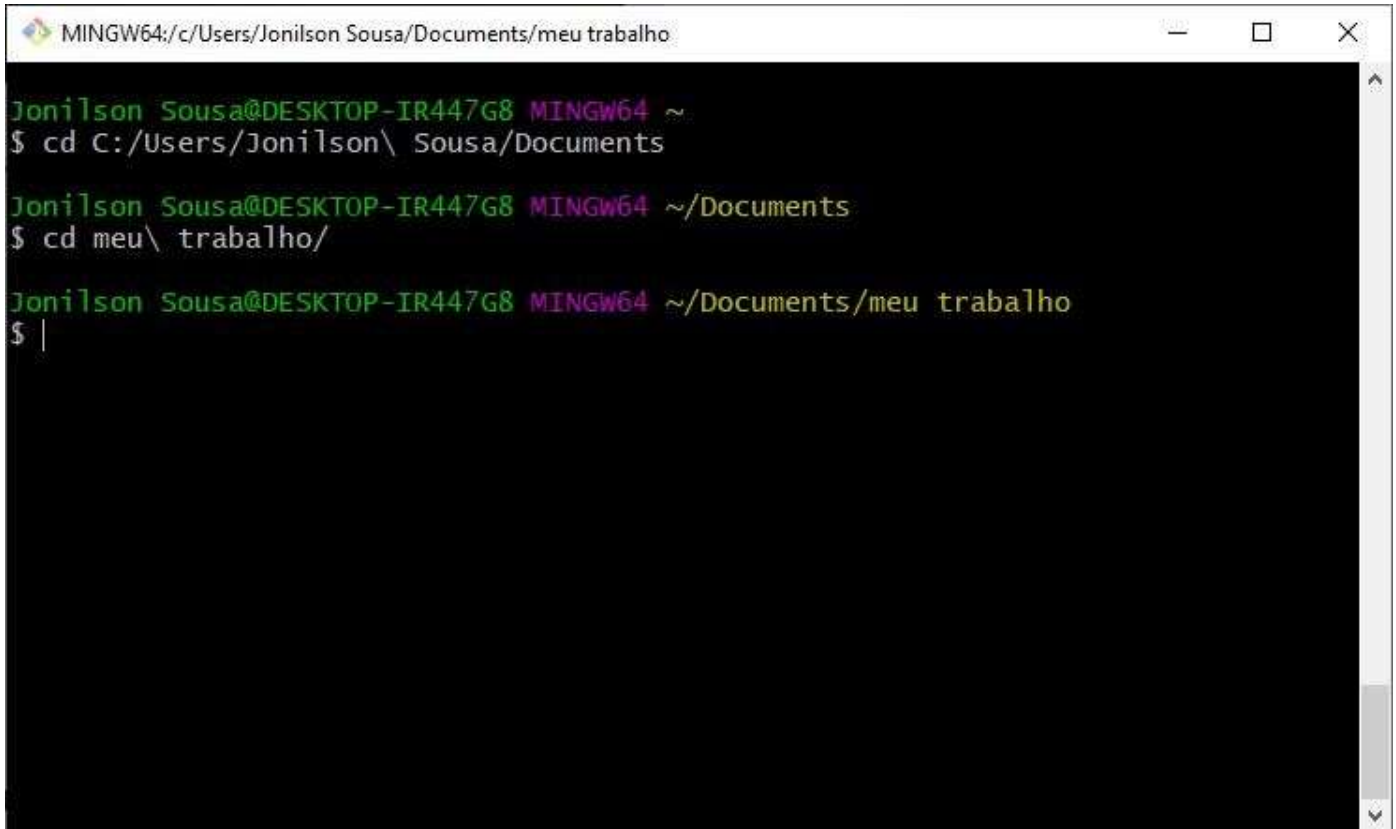
Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~/Documents
$ cd meu\ trabalho/

Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~/Documents/meu trabalho
$ |
```

Además de la utilización de las comillas, podemos utilizar la barra invertida en nombre de la carpeta. Observa el ejemplo:

```
cd C:/Users/Jonilson\ Sousa/Documents
```

```
cd meu\ trabalho
```

A screenshot of a Windows terminal window titled "MINGW64:/c/Users/Jonilson Sousa/Documents/meu trabalho". The terminal shows a series of commands and their outputs. The prompt is "Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~". The first command is "\$ cd C:/Users/Jonilson\ Sousa/Documents", which changes the directory to the user's Documents folder. The second command is "\$ cd meu\ trabalho/", which changes the directory to a subfolder named "meu trabalho". The third command is "\$", which shows the current directory as "~/Documents/meu trabalho". The terminal has a black background with green text for the prompt and pink text for the command. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
MINGW64:/c/Users/Jonilson Sousa/Documents/meu trabalho

Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~
$ cd C:/Users/Jonilson\ Sousa/Documents

Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~/Documents
$ cd meu\ trabalho/

Jonilson Sousa@DESKTOP-IR447G8 MINGW64 ~/Documents/meu trabalho
$ |
```

En los sistemas Linux

Las distribuciones Linux siguen en la misma línea: podemos utilizar las comillas (" ") o la barra invertida para acceder a las carpetas con nombres compuestos (con espacios):

```
jonilson@jonilson-Inspiron-7460: ~/Documentos/meu trabalho
Arquivo Editar Ver Pesquisar Terminal Ajuda
~/Documentos
> cd "meu trabalho"
~/Documentos/meu trabalho
>
```

```
jonilson@jonilson-Inspiron-7460: ~/Documentos/meu trabalho
Arquivo Editar Ver Pesquisar Terminal Ajuda
~/Documentos
> cd meu\ trabalho
~/Documentos/meu trabalho
>
```

Con base en los ejemplos vistos arriba, la preferencia por nombres sin espacios se vuelve interesante, ya que así podemos utilizar guion (-), guion bajo (_) o los nombres juntos (pienso que la comprensión queda un poco más difícil de esta forma), evitando, así, el uso de comillas o barras invertidas, que nos pueden costar en productividad.

Conclusión

Estos son algunos de los errores comunes que llevan a muchas personas a crear temas en el foro de Alura, entonces es importante entender el origen de estos errores y como corregirlos.

Eso trae beneficios en el desempeño cuando trabajamos en proyectos y cuando necesitamos utilizar servidores o máquinas sin interfaces gráficas, como es el caso de las máquinas creadas por **Vagrant** o entonces cuando trabajamos con contenedores **Docker**.

Puedes leer también:

- [Aprende a programar en 10 años](#)

- [6 malas prácticas que perjudican usuarios disléxicos](#)

ARTÍCULOS DE TECNOLOGÍA > DEVOPS

En Alura encontrarás variados cursos sobre DevOps. ¡Comienza ahora!

SEMESTRAL

US\$49,90

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 6 meses

¡QUIERO EMPEZAR A ESTUDIAR!

[Paga en moneda local en los siguientes países](#)

ANUAL

US\$79,90

un solo pago de US\$79,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 12 meses

¡QUIERO EMPEZAR A ESTUDIAR!

[Paga en moneda local en los siguientes países](#)

Acceso a todos
los cursos

Estudia las 24 horas,
dónde y cuándo quieras

Nuevos cursos
cada semana

NAVEGACIÓN

PLANES

INSTRUCTORES

BLOG

POLÍTICA DE PRIVACIDAD

TÉRMINOS DE USO

SOBRE NOSOTROS

PREGUNTAS FRECUENTES

¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

BLOG

PROGRAMACIÓN

FRONT END

DATA SCIENCE

INNOVACIÓN Y GESTIÓN

DEVOPS

AOVS Sistemas de Informática S.A
CNPJ 05.555.382/0001-33

SÍGUENOS EN NUESTRAS REDES SOCIALES



ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

POWERED BY

CURSOS

Cursos de Programación

Lógica de Programación | Java

Cursos de Front End

HTML y CSS | JavaScript | React

Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

Cursos de DevOps

Docker | Linux

Cursos de Innovación y Gestión

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics |
Liderazgo y Gestión de Equipos | Startups y Emprendimiento